

Willard Evan Bleick

MINIMAL-TIME SHIP ROUTING.

TA7
.U6
no.46

Library
U. S. Naval Postgraduate School
Monterey, California

Gaylord
PAMPHLET BINDER
Syracuse, N. Y.
Stockton, Calif.

UNITED STATES NAVAL POSTGRADUATE SCHOOL



MINIMAL-TIME SHIP ROUTING

by

W. E. Bleick and F. D. Faulkner

Professors of Mathematics and Mechanics

RESEARCH PAPER NO. 46

August 1964

Library

U. S. Naval Postgraduate School

Monterey, California

MINIMAL-TIME SHIP ROUTING

by

W. E. BLEICK and F. D. FAULKNER

Professors of Mathematics and Mechanics

Research Paper No. 46

UNITED STATES NAVAL POSTGRADUATE SCHOOL

Monterey, California

August 1964

Minimal-Time Ship Routing

W. E. BLEICK and F. D. FAULKNER

U. S. Naval Postgraduate School, Monterey, Calif.

ABSTRACT

A recent theory of minimal-time ship routing through time-dependent ocean wave height and direction fields is put to a numerical test by using a series of semidaily analyses furnished by the U. S. Navy Fleet Numerical Weather Facility. The interpolations and integrations required are found to be feasible. A resume of the theory is given.

1. Introduction

Haltiner, Hamilton and Arnason (1962) gave a relaxation method solution to the problem of minimal-time routing of ships through ocean wave height and direction fields dependent on the ship location coordinates only. The theory has been extended by Faulkner (1963) to the case where the wave height and direction depend on time also. The present paper confronts this time-dependent theory with actual wave height and direction analyses from the files of the U. S. Navy Fleet Numerical Weather Facility, and reports on practical problems which had to be solved in a test of the theory.

2. Polar velocity diagram

A basic ingredient of the theory is the polar diagram of Fig. 1, giving the ship velocity v as a function of the angle θ between the ship's heading and the wave direction. A diagram of this kind must be specified for each wave height H . The points L, M and N on the diagram correspond to the ship speed v_h in head waves, v_b in beam waves, and v_f in following waves. Empirical curves for these three speeds as functions of wave height H are available in the pioneer

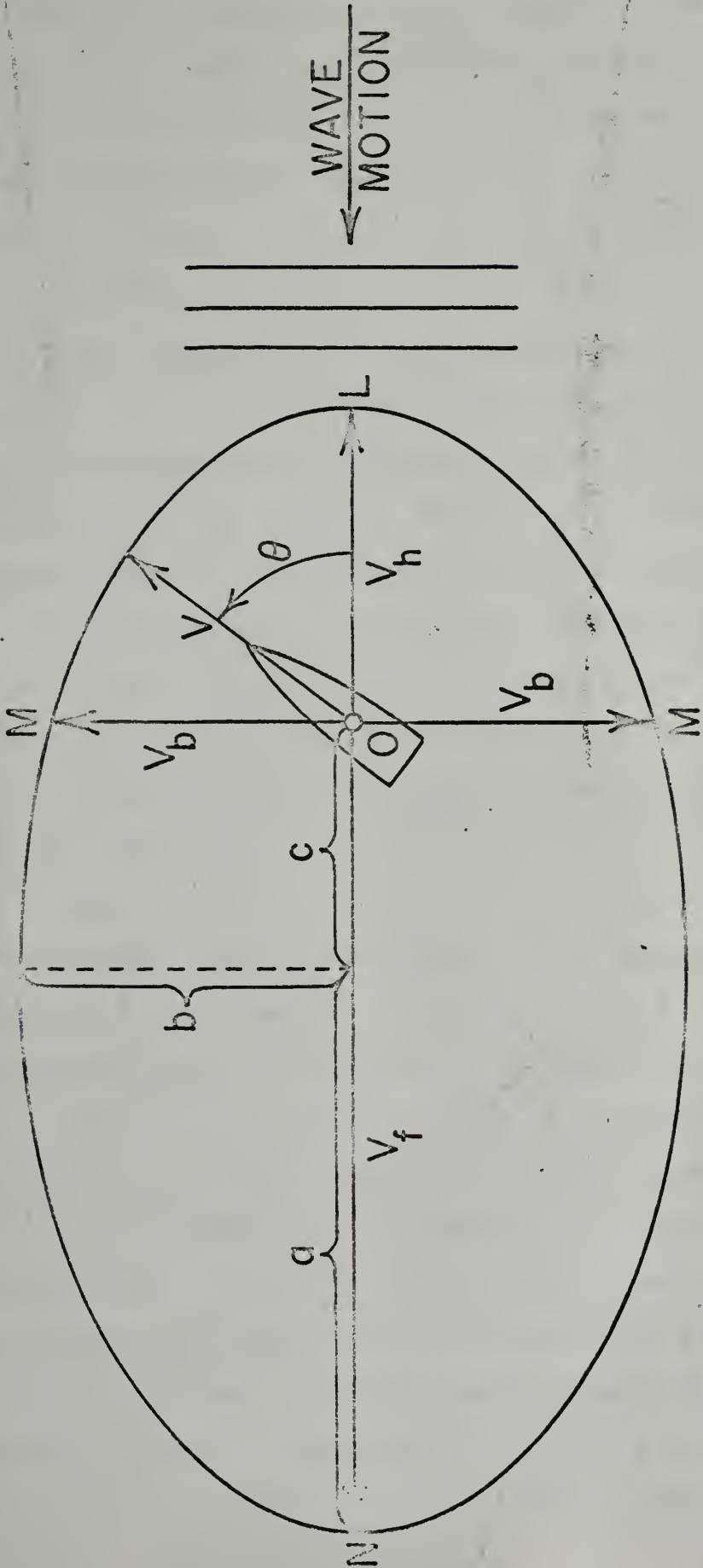


Fig. 1. Polar velocity diagram.

work of James (1957). His P2-S2-R2 ship type curves, shown in Fig. 2, have been chosen for use here. They have the appearance of being arcs of hyperbolas, at least approximately. This was confirmed when a least squares analysis showed that all three of v_h , v_b and v_f can be represented closely as functions of H by the hyperbolic arc

$$c_1 + c_2 H - [(c_3 + c_4 H)^2 + (c_1 - c_0)^2 - c_3^2]^{\frac{1}{2}} \quad (1)$$

where c_0 is the point common to all three speeds when $H=0$. The other four constants are related to the asymptotes $(c_2 + c_4)H + c_1 + c_3$ and $(c_2 - c_4)H + c_1 - c_3$. It was decided to construct the polar velocity diagram by fitting an ellipse to the points L, M and N. This resulted in semi-principal axes $a = (v_h + v_f)/2$, $b = a v_b / (v_h v_f)^{\frac{1}{2}}$, and a distance to the eccentric pole 0 given by $c = (v_f - v_h)/2$. Note that the pole 0 is not to be construed as a focus of the ellipse. A further least squares analysis showed that the semiaxes a and b are closely representable also by hyperbolic functions of the form of (1), but not c which must be calculated as $c = (v_f - v_h)/2$.

3. Coordinate system

The ocean wave height and direction data of the semidaily Fleet Numerical Weather Facility analyses are presented in a south-polar stereographic projection of the northern hemisphere upon a plane passing through the circle of 60° North latitude. A rectangular coordinate system is set up in this projecting plane with the Ox and Oy axes parallel to the projections of the meridians of 10° and 100° East longitude respectively. A 62 by 62 grid is constructed using these axes with $x=y=31$ defining the projection of the North pole. The mesh distance between grid lines corresponds to a distance of 381 km at 60° North latitude where the projection is true. The radius of the equator's projection is 31.205 mesh units. The map scale factor m,

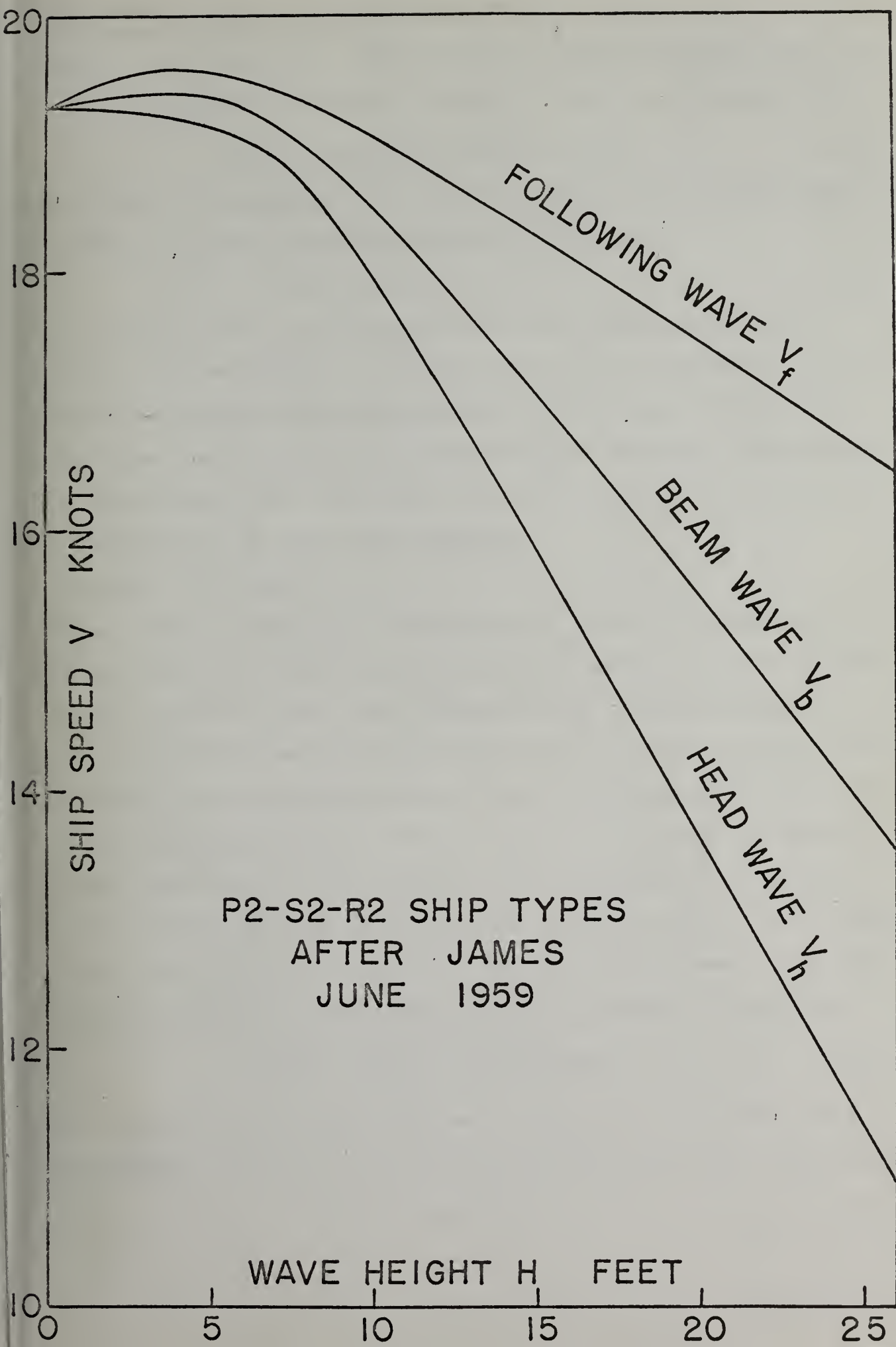


Fig. 2. Ship speed in following, beam and head waves.

defined as the ratio of a differential distance in the Oxy plane to the corresponding differential distance on the earth's surface, is

$$m = [973.75 + (x-31)^2 + (y-31)^2] / 1043.6 \quad (2)$$

Let x, y be the coordinates of a ship's projection in the Oxy plane at time t . Then the projected speed of the ship is

$$V(x, y, t, \theta) = v(H, \theta) m \quad (3)$$

where $v(H, \theta)$ is the actual geographical ship speed of the polar velocity diagram of Fig. 1, and where $H(x, y, t)$ is obtained by interpolation in the Fleet Numerical Weather Facility grid wave height data. Since the stereographic projection is a conformal transformation preserving angles and their senses, the angle θ is the same in the Oxy plane as on the earth's surface.

4. Resume of the theory

Fig. 3 shows a ship at the point (x, y, t) in the stereographic projection plane on a route from fixed initial point A at $t=0$ to fixed terminal point B at $t=T$. The elliptical polar velocity diagram for $V=mv$ is plotted at this point by interpolation in the semidaily wave height $H(x, y, t)$ and wave direction $K(x, y, t)$ grid values of the Fleet Numerical Weather Facility. The special type of interpolation required is discussed in the Appendix. The direction of the ship's velocity vector $\underline{V} = i\dot{x} + j\dot{y}$ is the control angle p , which is to be chosen at each point so as to minimize the transit time T from A to B. The equations of motion of the ship's projection in the Oxy plane are

$$\varphi_1 = \dot{x} - V \cos p = 0, \quad \varphi_2 = \dot{y} - V \sin p = 0 \quad (4)$$

where $V = |\underline{V}| = V(x, y, t, p)$. The problem of minimizing T is equivalent to the Lagrange calculus of variations problem of requiring the integral

$$I = \int_0^T (1 + \lambda \varphi_1 + \mu \varphi_2) dt \quad (5)$$

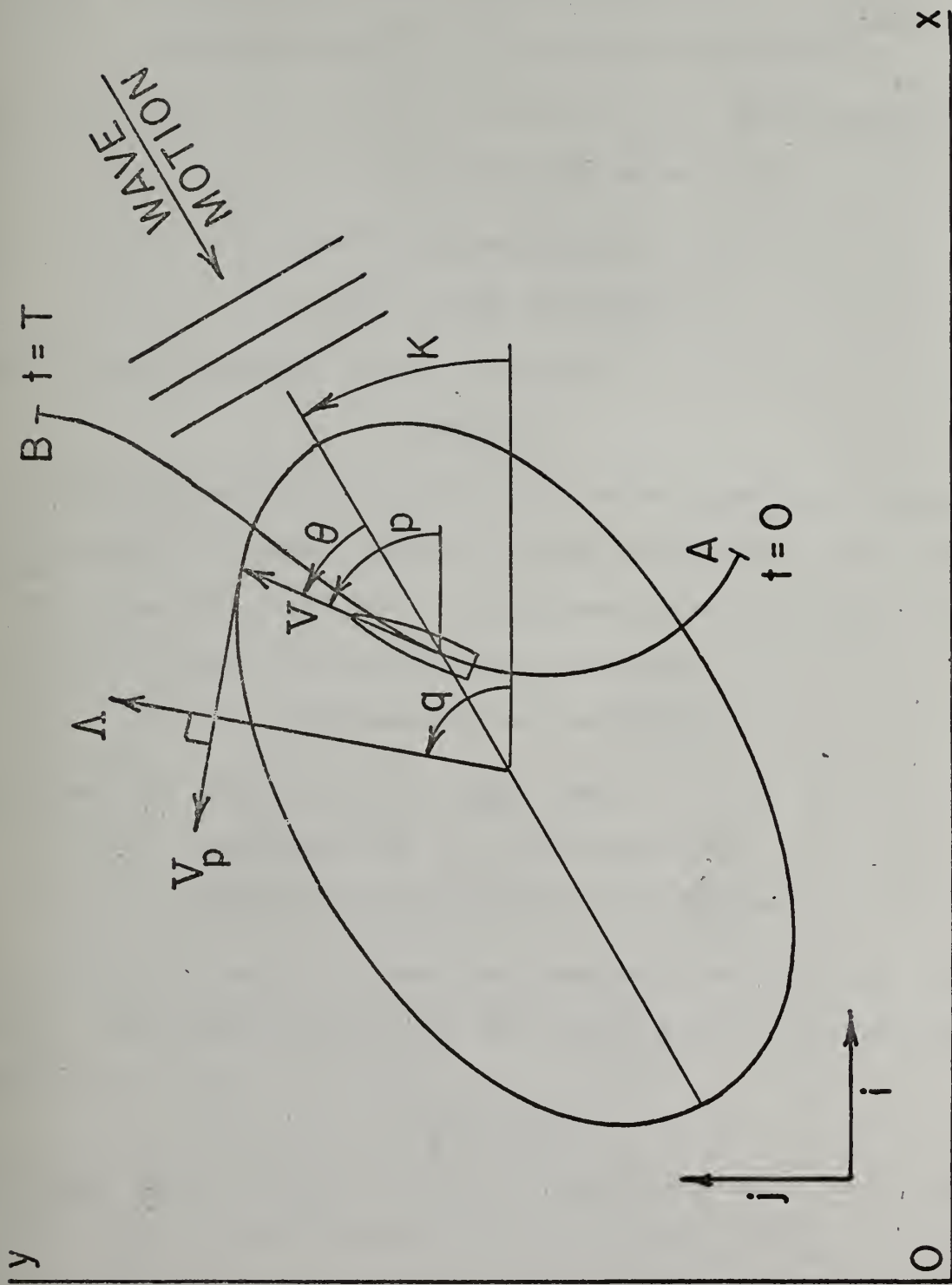


Fig. 3. Ship motion in stereographic Oxy plane.

to be stationary, where $\lambda(t)$ and $\mu(t)$ are continua of Lagrangian multipliers. Let the time at the fixed terminal point B be varied to $T+\Delta T$. The vanishing first variation of I is

$$\delta I = \Delta T + [\lambda \delta x + \mu \delta y]_0^T - \int_0^T (\varphi_3 \delta x + \varphi_4 \delta y + \varphi_5 \delta p) dt = 0. \quad (6)$$

The coefficients of $\delta x, \delta y, \delta p$ in $\delta I = 0$ give the Euler equations (7), (8) and (9), consisting of the adjoint equations

$$\varphi_3 = \dot{\lambda} + (\lambda \cos p + \mu \sin p) V_x = 0, \quad (7)$$

$$\varphi_4 = \dot{\mu} + (\lambda \cos p + \mu \sin p) V_y = 0, \quad (8)$$

and the scalar product control equation

$$\varphi_5 = \hat{\Lambda} \cdot \underline{V}_p = 0, \quad (9)$$

where the adjoint vector $\hat{\Lambda} = i\lambda + j\mu$, and where $\underline{V}_p = \partial V / \partial p = i(V_p \cos p - V \sin p) + j(V_p \sin p + V \cos p)$ is the tangent vector to the polar velocity diagram of Fig.3. Eq. (9) implies the orthogonality of $\hat{\Lambda}$ and \underline{V}_p as shown in Fig.3. Eq.(9) may be written also in the form

$$p = \arctan(\mu/\lambda) + \arctan(V_p/V). \quad (10)$$

The fixed end points A and B imply that

$$dx(0) = \delta x(0) = 0, \quad dy(0) = \delta y(0) = 0, \quad (11)$$

$$dx(T) = (\dot{x}\Delta t + \delta x)_T = 0, \quad dy(T) = (\dot{y}\Delta t + \delta y)_T = 0. \quad (12)$$

Use of Eqs.(11) and (12) makes the remaining terms of (6) proportional to ΔT , whose coefficient gives the scalar product transversality condition

$$(\hat{\Lambda} \cdot \underline{V})_T = 1 > 0 \quad (13)$$

meaningful for sign only because of the homogeneity of (7) and (8). Eq.(13) implies that the angle $(q-p)$ between \underline{V} and $\hat{\Lambda}$ is acute, as shown in Fig.3. A further implication of (10) and (13) is that the quadrant of p is such that

$$\cos p = (\lambda V - \mu V_p) / \hat{\Lambda} R, \quad \sin p = (\lambda V_p + \mu V) / \hat{\Lambda} R, \quad (14)$$

where $\Lambda = |\Lambda| = (\lambda^2 + \mu^2)^{\frac{1}{2}}$ and $R = |V_p| = (V^2 + V_p^2)^{\frac{1}{2}}$.

The simultaneous numerical integration of (4), (7), (8) and (14) is carried out together with a Newton-Raphson iteration as follows: Let λ_1, μ_1 and λ_2, μ_2 be two linearly independent solutions of the adjoint Eqs. (7) and (8) corresponding to the columns of the matrix

$$\underline{E}(t) = \begin{bmatrix} \lambda_1 & \lambda_2 \\ \mu_1 & \mu_2 \end{bmatrix} \quad (15)$$

where $\underline{E}(0) = \underline{I}$ is the identity matrix. The λ, μ of (14) are taken as the linear combinations $\lambda = \lambda_1 \cos \alpha + \lambda_2 \sin \alpha$ and $\mu = \mu_1 \cos \alpha + \mu_2 \sin \alpha$. The variation δp is found by total differentiation of (10) to be

$$\delta p = R |\underline{E}| \delta \alpha / \Lambda^2 (V^2 + 2V_p^2 - V V_{pp}) \quad (16)$$

where $|\underline{E}|$ is the determinant of \underline{E} . Assume that a solution of the ship motion Eqs. (4) has been found, corresponding to (7), (8) and (14) for some value of α , which falls short of the fixed end point B at $t=T$ by the coordinate differences $\Delta x(T)$ and $\Delta y(T)$. Using this solution and holding T fixed, find the variation of the vanishing matrix integral

$$\int_0^T [\varphi_1, \varphi_2] \underline{E}(t) dt = 0. \quad (17)$$

Since the columns of $\underline{E}(t)$ satisfy the adjoint Eqs. (7) and (8), one obtains the 1×2 matrix equation

$$[\delta x, \delta y]_T \underline{E}(T) = \int_0^T [(V_p \cos p - V \sin p), (V_p \sin p + V \cos p)] \underline{E} \delta p dt. \quad (18)$$

Substitution from (14) and (16) into (18) gives

$$[\delta x, \delta y]_T = [-\mu, \lambda]_T J \delta \alpha \quad (19)$$

where

$$J = - \frac{1}{|\underline{E}(T)|} \int_0^T R^3 |\underline{E}|^2 dt / \Lambda^3 (V^2 + 2V_p^2 - V V_{pp}). \quad (20)$$

Now vary the terminal time from T to $T + \Delta T$ and substitute

$$[\delta x, \delta y]_T = [\Delta x, \Delta y]_T - [\dot{x}, \dot{y}]_T \Delta T \quad (21)$$

into (19) to obtain the Newton-Raphson equations

$$\begin{aligned} \dot{x}(T)\Delta T - J\mu(T)\delta\alpha &= \Delta x(T) \\ \dot{y}(T)\Delta T + J\lambda(T)\delta\alpha &= \Delta y(T) \end{aligned} \tag{22}$$

for the determination of ΔT and $\delta\alpha$ on a varied trajectory which attempts to correct the errors $\Delta x(T)$ and $\Delta y(T)$. The iteration to successive varied trajectories is continued until the terminal errors are acceptable. A suitable initial guess for the angle α is the inclination angle of the straight line from A to B.

5. Numerical example

Ten successive semidaily analyses of wave height and direction, starting at 06Z on 4 May 1963, were furnished by the Fleet Numerical Weather Facility. The maximum speed of the chosen P2-S2-R2 ship type is 19.6 knots. This combination of data precluded a trip of great length. It was decided to select an area of continued extreme wave height for the example. Such an area was found centering at 30° North latitude and 162° East longitude. Figs. 4, 5 and 6 show two computed minimal-time ship tracks in the area, with contours of wave height in feet and wave direction arrows. The arc traversed by the ship during the 6 hours preceding and/or the 6 hours following the time of each Figure is shown as a dashed curve. The minimal-time track AB required 2.488 days with a 3.0% saving over the geodesic track. The minimal-time track CD required 2.656 days with a 1.3% saving over the geodesic track. The severity of the sea conditions in the area preclude any more spectacular saving. The highly non-analytic nature of the wave height in the area was found to affect the convergence of the Newton-Raphson iteration of (22). It was found necessary to halve the values of ΔT and $\delta\alpha$ in order to avoid a divergent oscillation. Resort to this stratagem was found to be unneces-

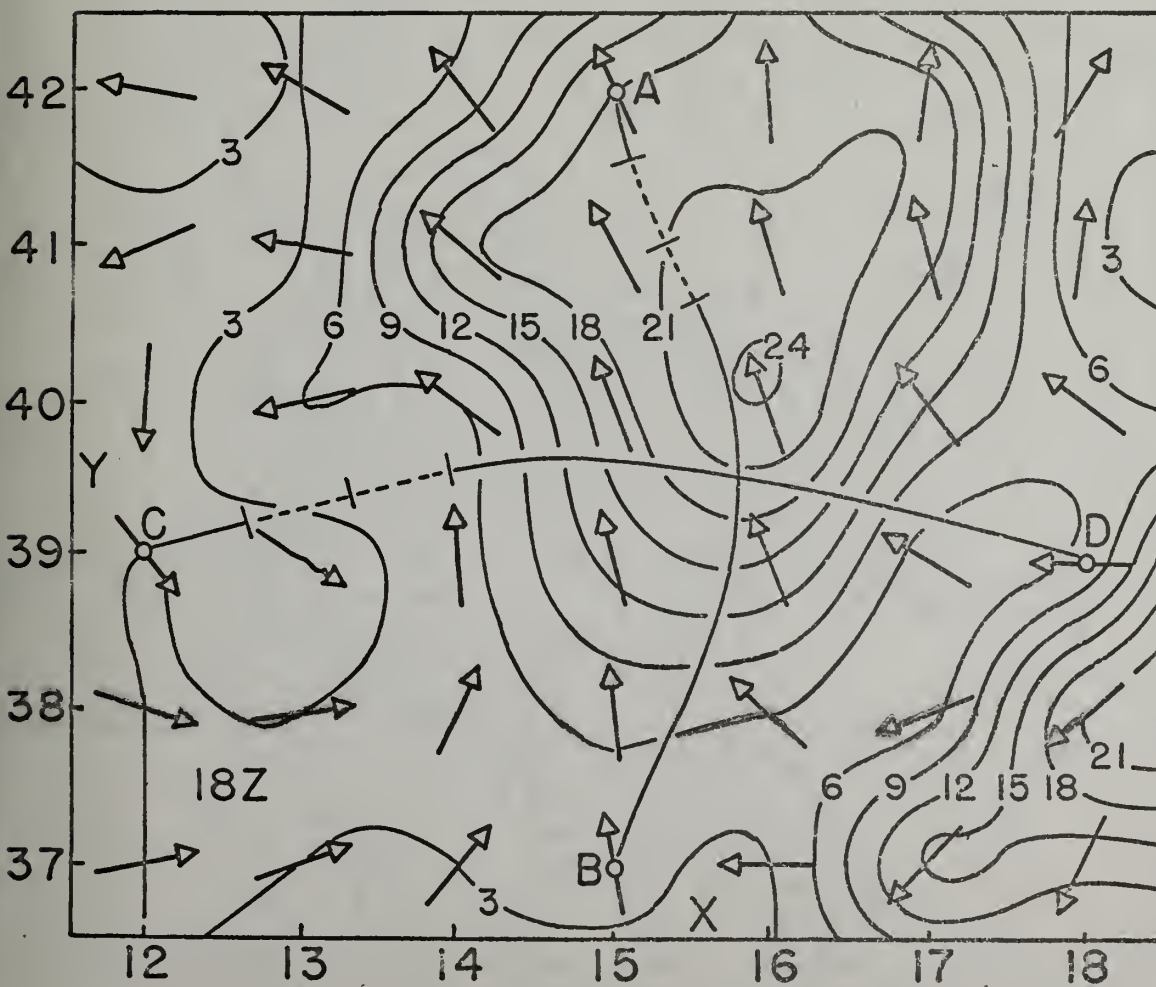
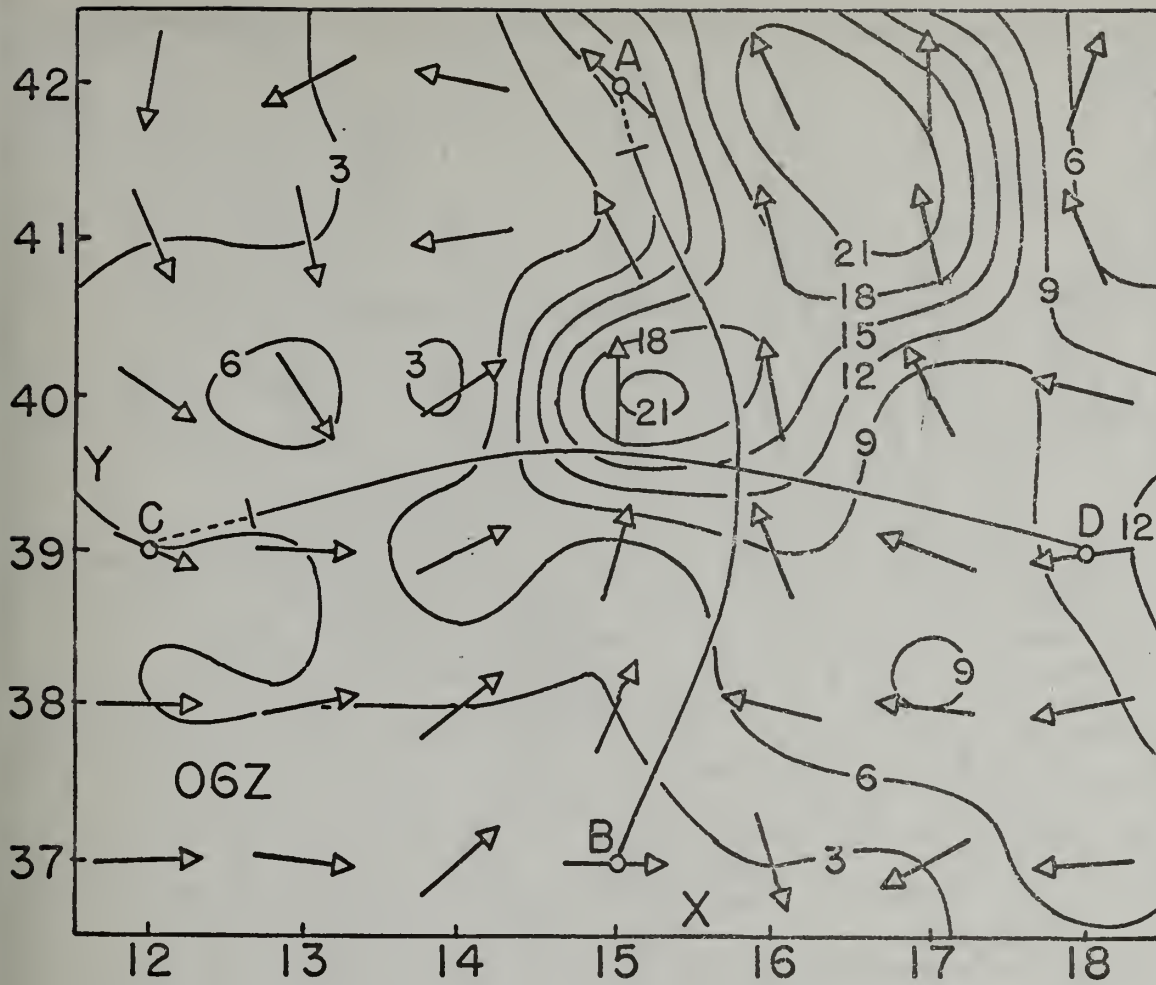


Fig. 4. Sea conditions at 06Z and 18Z on 4 May 1963.

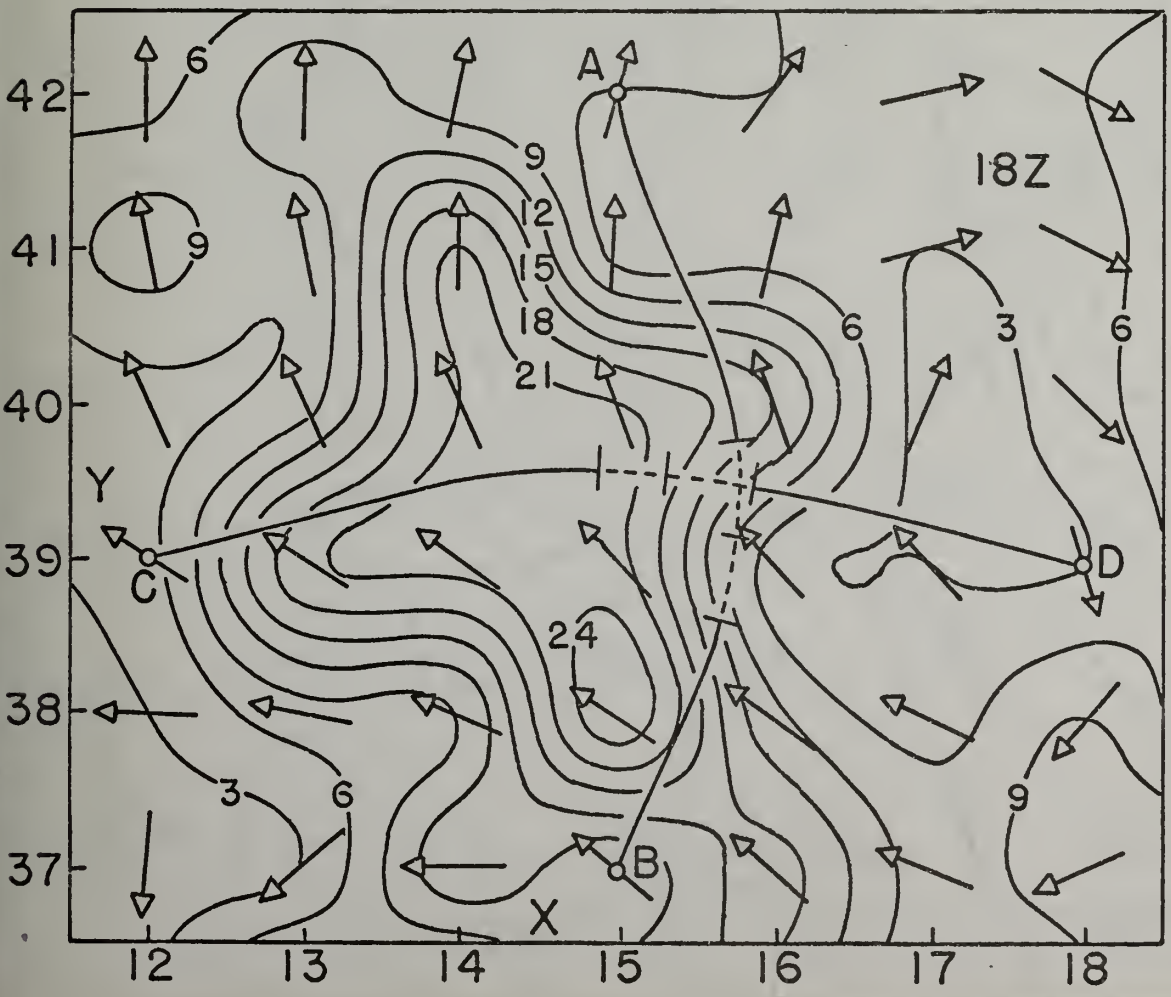
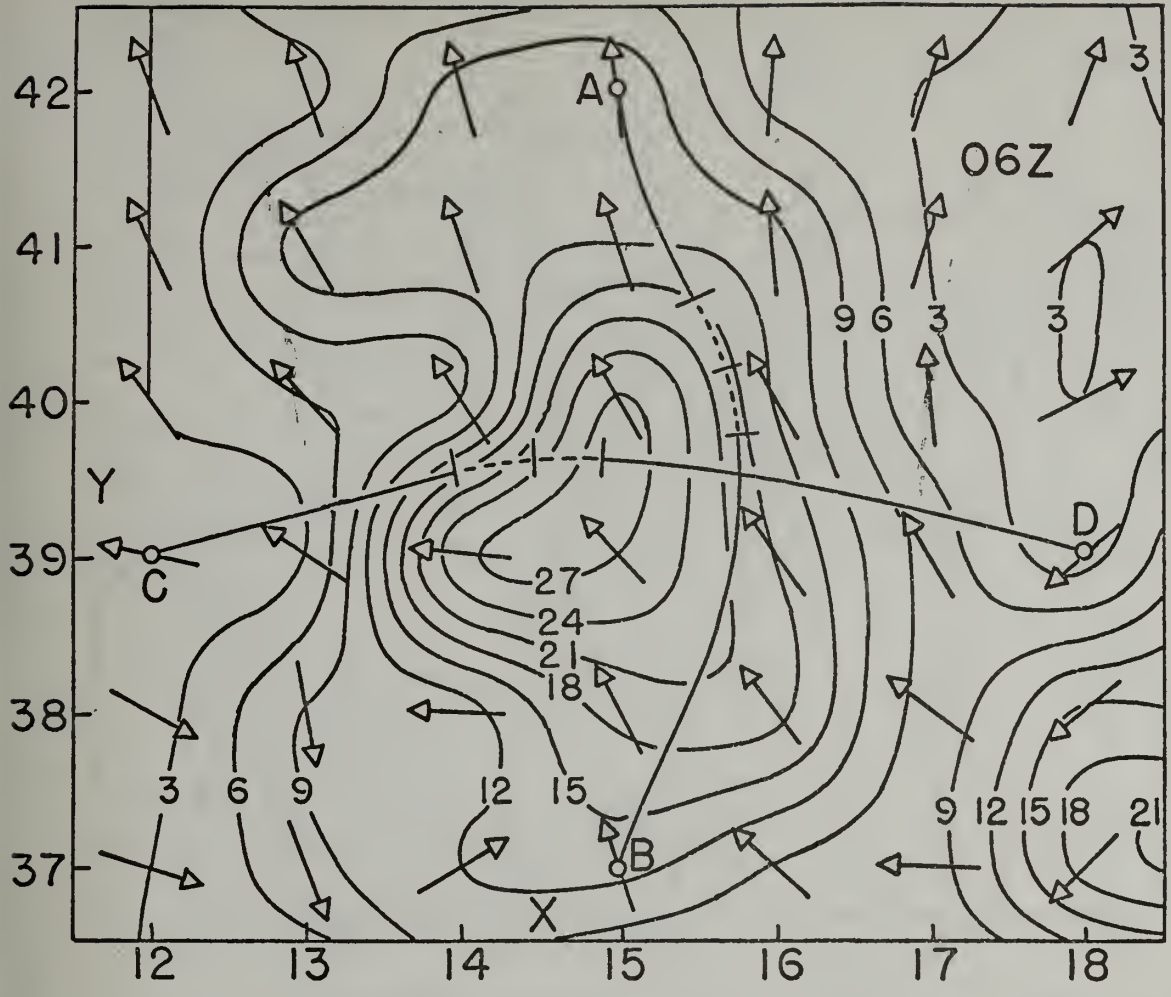


Fig. 5. Sea conditions at 06Z and 18Z on 5 May 1963.

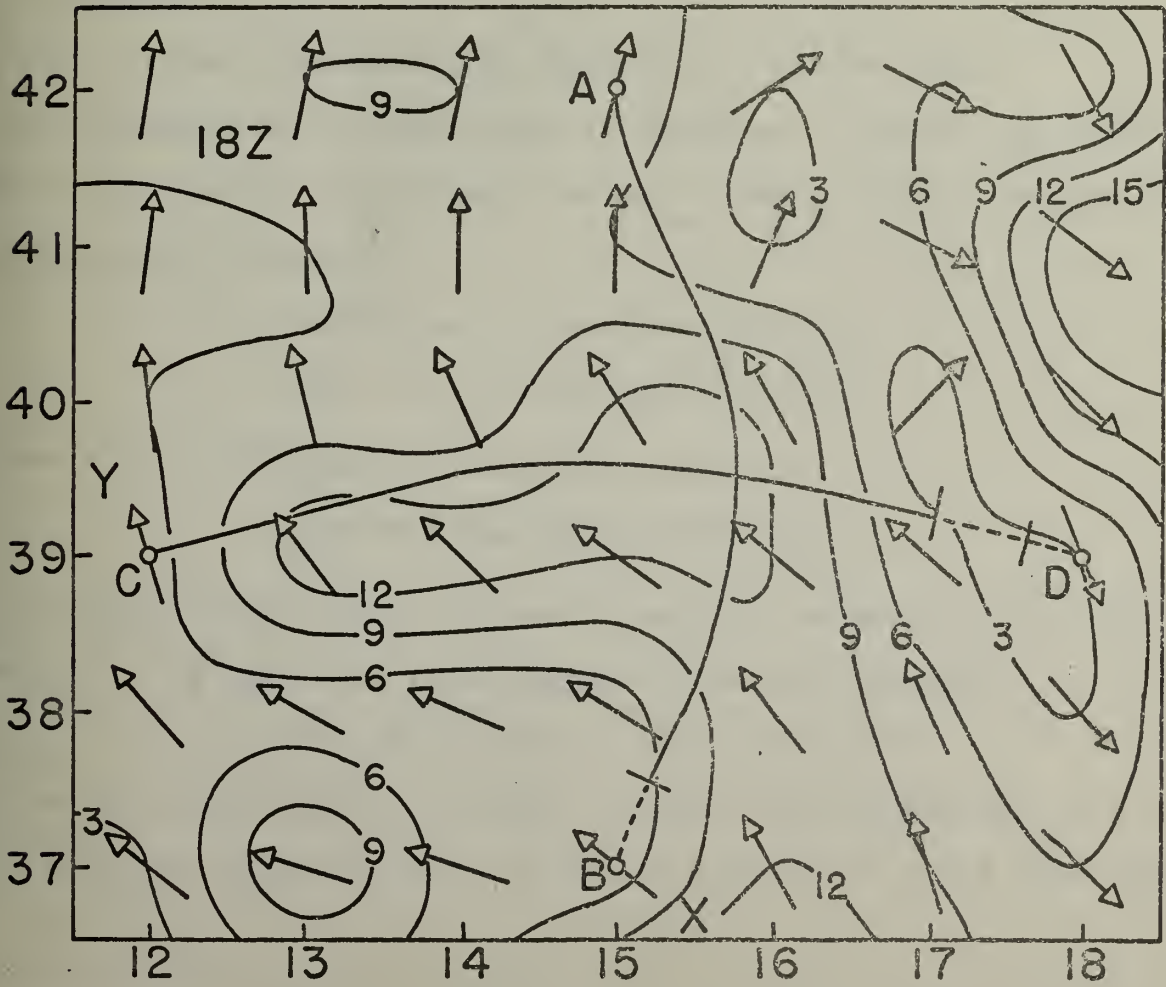
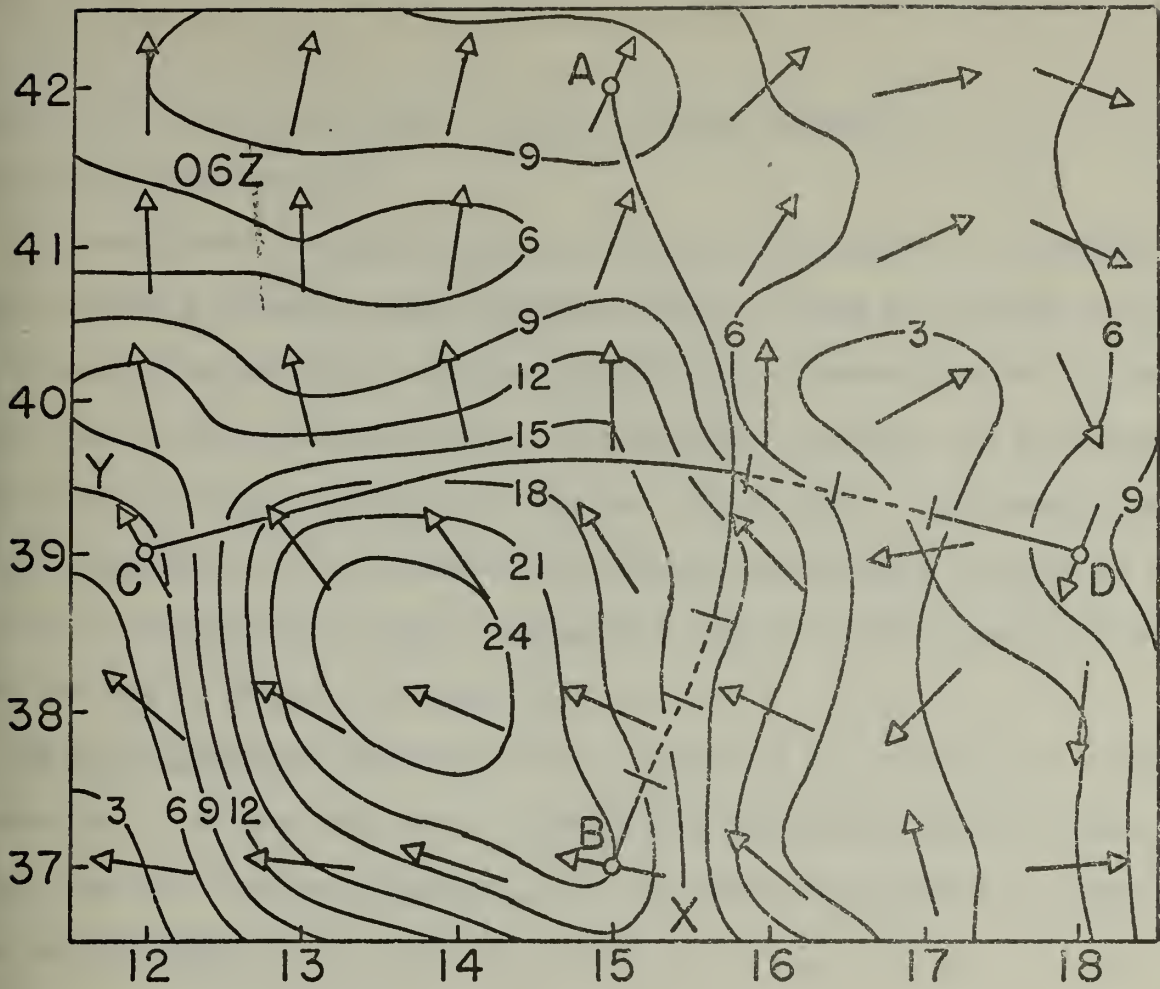


Fig. 6. Sea conditions at 06Z and 18Z on 6 May 1963.

sary when the wave height was more nearly analytic.

6. Concluding remarks

The numerical integrations involved in the theory of minimal-time ship routing through time-dependent wave fields are found to be feasible. The necessary three-dimensional interpolations in the wave field data, discussed in the Appendix, present no problem. Convergence problems may arise, but can be solved by the described delayed approach to the limit. The authors can supply copies of their Fortran programs for ship routing and for the cubic-interpolation contouring of Figs. 4, 5 and 6.

Acknowledgements. This work was supported by the Office of Naval Research. Aid from the U. S. Navy Fleet Numerical Weather Facility and from the Computer Facility, U. S. Naval Postgraduate School, is acknowledged.

7. Appendix

Some pertinent mathematical details are listed here.

The geographic wave direction ψ , measured clockwise from the North, must be converted to the unit vector $i \cos K + j \sin K$ in the stereographic grid system by

$$\begin{aligned}\cos K &= -[(x-31)\cos\psi + (y-31)\sin\psi]/r \\ \sin K &= [(x-31)\sin\psi - (y-31)\cos\psi]/r\end{aligned}\tag{23}$$

where $r^2 = (x-31)^2 + (y-31)^2$. Then the derivative

$$K_x = \cos K (\partial \sin K / \partial x) - \sin K (\partial \cos K / \partial x).\tag{24}$$

The derivatives $V_x = mv_x + vm_x$ and $V_p = mv_p$ are obtained most conveniently by the implicit differentiation of the equation

$$[v \sin(p-K)/b]^2 + [(c+v \cos(p-K))/a]^2 = 1\tag{25}$$

of the elliptical polar velocity diagram, and noting that a, b, c are functions of $H(x, y, t)$, and that K depends on x, y, t . The com-

plexity of the result is reduced by introducing the parameter β defined by

$$\begin{aligned} \sin\beta &= b \sin(q-K)/s = v \sin(p-K)/b \\ \cos\beta &= a \cos(q-K)/s = [v \cos(p-K) + c]/a \end{aligned} \quad (26)$$

where $s^2 = a^2 \cos^2(q-K) + b^2 \sin^2(q-K)$.

The numerical integration of the adjoint Eqs.(7) and (8) demands an interpolation formula for $H(x,y,t)$, $\cos K(x,y,t)$ and $\sin K(x,y,t)$ which guarantees the continuity of these functions and of their first space and time derivatives where any of x,y,t assume grid values. A 16-point interpolation formula to accomplish this is obtained from the 4×4 matrix F , whose four rows and columns of function entries correspond to four successive x and y grid values respectively. The interpolation mesh cell is the central cell of the array, with x and y measured from the cell center, and with the mesh distance considered to be two units. The formula is

$$F(x,y) = P(x) F P'(y) / 256 \quad (27)$$

where the matrix

$$P(x) = [(1-x)(x^2-1), (x-1)(3x^2+2x-9), (x+1)(9+2x-3x^2), (x+1)(x^2-1)] \quad (28)$$

and the prime indicates matrix transposition. Interpolation in the time dimension is accomplished by the similar 2-unit-mesh central-difference formula

$$F(t) = [F(-3), F(-1), F(1), F(3)] P'(t) / 16 \quad (29)$$

which guarantees the continuity of $F(t)$ and dF/dt at each end of the central time interpolation mesh. This formula is consistent with parabolic interpolation at the beginning or end of a time series, where central differences are not available. An interpolated vector $i \cos K(x,y,t) + j \sin K(x,y,t)$ should be normalized before use.

REFERENCES

- Haltiner, G. J., Hamilton, H. D. and Arnason, G., 1962: Minimal-time ship routing. J. Applied Meteor., 1, 1-7.
- Faulkner, F. D., 1963: Numerical methods for determining optimum ship routes. Navigation, 10, 351-367.
- James, R. W., 1957 (revised 1959): Application of wave forecasts to marine navigation. U. S. Navy Hydrographic Office.

FIGURE LEGENDS

- Fig. 1. Polar velocity diagram.
- Fig. 2. Ship speed in following, beam and head waves.
- Fig. 3. Ship motion in stereographic Oxy plane.
- Fig. 4. Sea conditions at 06Z and 18Z on 4 May 1963.
- Fig. 5. Sea conditions at 06Z and 18Z on 5 May 1963.
- Fig. 6. Sea conditions at 06Z and 18Z on 6 May 1963.


```

PROGRAM MINVOY
YVARS(1)=XLM1  YVARS(2)=XMU1  YVARS(3)=XLM2  YVARS(4)=XMU2  JJ=0  EW
YVARS(5)=X      YVARS(6)=Y      YVARS(7)=XJ    YVARS(8)=S      JJ=1  NS
DIMENSION KH(19,12,10),HT(19,12,10),KK(19,12,10),COSK(19,12,10),
+ SINK(19,12,10),YVARS(8),DY(8),YC(8),C(4),TAU(300),X(300),Y(300),
+ CAPLAM(300),PP(300),QQ(300),WK(300),VS(300),WH(300),XJ(300),
+ S(300),AK(4,8)
EQUIVALENCE (KH,HT), (KK,COSK)
READ 1, KH,KK, JJ, LMAX, KXST, KYST, KXFN, KYFN, ALF, T, FAC, FMUL
1 FORMAT ( 60(38I2/), 60(38I2/), 6I3, 4F12.9 )
PRINT 2, JJ, LMAX, KXST, KYST, KXFN, KYFN, FAC, FMUL
2 FORMAT (1H0, 6I3, 2F12.9)
DO 4 I=1,19
DELI = I-26
DO 4 J=1,12
DELJ = J+1
ROOT = SQRTF(DELI*DELI + DELJ*DELJ)
DO 4 K=1,10
HT(I, J, K) = KH(I, J, K)
ANGLE = KK(I, J, K) * 10
ANGLE = ANGLE/57.2957 7951
COS = COSF(ANGLE)
SIN = SINF(ANGLE)
4 COSK(I, J, K) = -(DELI*COS + DELJ*SIN)/ROOT
SINK(I, J, K) = (DELI*SIN - DELJ*COS)/ROOT
C(1) = 0.0
C(2) = 0.5
C(3) = 0.5
C(4) = 1.0
X(1) = KXST
Y(1) = KYST
TAU(1) = 0.0
XFIN = KXFN
YFIN = KYFN
WH(1) = HT(KXST, KYST, 1)
CALL POLAR (COSK(KXST, KYST, 1), SINK(KXST, KYST, 1), WK(1))
XSTEP = T/FAC
CALL ABC (WH(1), AI, BI, CI, DAI, DBI, DCI)
CAPLAM(1) = 1.0
S(1) = 0.0
XJ(1) = 0.0
DO 18 L=1, LMAX
Q = ALF
QQ(1) = Q * 57.2957 7951
COSA = COSF(ALF)
SINA = SINF(ALF)
COSQ = COSA
SINQ = SINA
COSQMK = COSQ*COSK(KXST, KYST, 1) + SINQ*SINK(KXST, KYST, 1)
SINQMK = SINQ*COSK(KXST, KYST, 1) - COSQ*SINK(KXST, KYST, 1)
ABS = AI*COSQMK
ORD = BI*SINQMK
HYP = SQRTF(ABS*ABS + ORD*ORD)
SINB = ORD/HYP
COSB = ABS/HYP
VMAJ = AI*COSB - CI
VMIN = BI*SINB
VS(1) = SQRTF(VMAJ*VMAJ + VMIN*VMIN)
CALL POLAR (VMAJ, VMIN, PMK)
PP(1) = PMK + WK(1)
XVAR = 0.0
DO 5 I=1, 8
5 YVARS(I) = 0.0
YVARS(1) = 1.0
YVARS(4) = 1.0
YVARS(5) = X(1)
YVARS(6) = Y(1)
N1 = T/XSTEP + 1.0
XN1 = N1
STEP = T/XN1
N2 = N1 + 1
DO 14 K=2, N2
DO 7 I=1, 4
XC = XVAR + C(I)*STEP
DO 6 J=1, 8
6 YC(J) = YVARS(J) + C(I)*AK(I-1, J)
XLAM = YC(1)*COSA + YC(3)*SINA
XMU = YC(2)*COSA + YC(4)*SINA
CLAM = SQRTF(XLAM*XLAM + XMU*XMU)
CALL TERP (HT, YC(5), YC(6), XC, H, HX, HY)

```



```

CALL ABC (H,FA,FB,FC,DA,DB,DC)
CALL TERP (COSK,YC(5),YC(6),XC,CK,CKX,CKY)
CALL TERP (SINK,YC(5),YC(6),XC,SK,SKX,SKY)
ROOT = SQRTF(CK*CK + SK*SK)
CK = CK/ROOT
SK = SK/ROOT
DKX = CK*SKX - SK*CKX
DKY = CK*SKY - SK*CKY
COSQ = XLAM/CLAM
SINQ = XMU/CLAM
COSQMK = COSQ*CK + SINQ*SK
SINQMK = SINQ*CK - COSQ*SK
ABS = FA*COSQMK
ORD = FB*SINQMK
HYP = SQRTF(ABS*ABS + ORD*ORD)
SINB = ORD/HYP
COSB = ABS/HYP
VMAJ = FA*COSB - FC
VMIN = FB*SINB
V = SQRTF(VMAJ*VMAJ + VMIN*VMIN)
COSP = (CK*VMAJ - SK*VMIN)/V
SINP = (SK*VMAJ + CK*VMIN)/V
COST = VMAJ/V
VBR = V/18.702 181 818
DELX = YC(5) - 26.0
DELY = YC(6) + 1.0
EMFI = (973.75 + DELX*DELX + DELY*DELY)/1043.638 743
CAPV = VBR*EMFI
DY(5) = CAPV*COSP
DY(6) = CAPV*SINP
EMFIX = DELX/521.819 3715
EMFIY = DELY/521.819 3715
DY(8) = CAPV
B2MA2 = FB*FB - FA*FA
AMCCB = FA - FC*COSB
RAT = SINB*(FA*FC + B2MA2*COSB)/(FB*AMCCB)
VP = RAT * V
CAPVP = RAT * CAPV
DIV = 1. + RAT**2 - (V/AMCCB)**2 * ((VP*COST/SINB) - (B2MA2*SINB**2/FB))/FB
DET = YC(1)*YC(4) - YC(2)*YC(3)
RADIX = SQRTF(1.0 + RAT*RAT)
DY(7) = (CAPV*DET*DET/DIV) * (RADIX/CLAM)**3
FNUM = -RAT*AMCCB
DBDH = DA*COSB**2 + FA*DB*SINB**2/FB - DC*COSB
RATX = (FNUM*DKX + HX*DBDH)/AMCCB
RATY = (FNUM*DKY + HY*DBDH)/AMCCB
VBRX = RATX*VBR
VBRY = RATY*VBR
CAPVX = VBRX*EMFI + VBR*EMFIX
CAPVY = VBRY*EMFI + VBR*EMFIY
DY(1) = -CAPVX*(YC(1)*COSP + YC(2)*SINP)
DY(2) = -CAPVY*(YC(1)*COSP + YC(2)*SINP)
DY(3) = -CAPVX*(YC(3)*COSP + YC(4)*SINP)
DY(4) = -CAPVY*(YC(3)*COSP + YC(4)*SINP)
DO 7 J=1,8
7 AK(I,J) = STEP * DY(J)
DO 8 J=1,8
8 YVARS(J) = YVARS(J) + ( AK(1,J)+2.*AK(2,J)+2.*AK(3,J)+AK(4,J) )/6.
XVAR = XVAR + STEP
TAU(K) = TAU(K-1) + STEP
X(K) = YVARS(5)
Y(K) = YVARS(6)
IF (N2-K) 80,81,80
IF (LMAX-L) 82,81,82
80 XLAM = YVARS(1)*COSA + YVARS(3)*SINA
81 XMU = YVARS(2)*COSA + YVARS(4)*SINA
CLAM = SQRTF(XLAM*XLAM + XMU*XMU)
CAPLAM(K) = CLAM
CALL TERP (HT,X(K),Y(K),TAU(K),WH(K),HX,HY)
CALL ABC (WH(K),FA,FB,FC,DA,DB,DC)
CALL TERP (COSK,X(K),Y(K),TAU(K),CK,CKX,CKY)
CALL TERP (SINK,X(K),Y(K),TAU(K),SK,SKX,SKY)
ROOT = SQRTF(CK*CK + SK*SK)
CK = CK/ROOT
SK = SK/ROOT
COSQ = XLAM/CLAM
SINQ = XMU/CLAM
COSQMK = COSQ*CK + SINQ*SK
SINQMK = SINQ*CK - COSQ*SK
ABS = FA*COSQMK
ORD = FB*SINQMK

```

81
82
83
84
85
86
87
88
89
90
91
92
93
94
95
96
97
98
99
100
101
102
103
104
105
106
107
108
109
110
111
112
113
114
115
116
117
118
119
120
121
121.5
122
123
124
125
126
127
128
129
130
131
132
133
134
135
136
137
138
139
140
141
141.0
141.1
142
143
144
145
146
147
148
149
150
151
152
153
154
155
156
157
158


```

HYP = SQRTF(ABS*ABS + ORD*ORD)
SINB = ORD/HYP
COSB = ABS/HYP
VMAJ = FA*COSB - FC
VMIN = FB*SINB
VS(K) = SQRTF(VMAJ*VMAJ + VMIN*VMIN)
COSP = (CK*VMAJ - SK*VMIN)/VS(K)
SINP = (SK*VMAJ + CK*VMIN)/VS(K)
CALL POLAR (COSP, SINP, PP(K))
CALL POLAR (COSQ, SINQ, QQ(K))
CALL POLAR (CK, SK, WK(K))
XJ(K) = YVARS(7)/(YVARS(1)*YVARS(4) - YVARS(2)*YVARS(3))
S(K) = YVARS(8)
IF (N2-K) 82,15,82
IF (JJ) 88,83,88
IF (XFIN-X(1)) 85,21,84
IF (X(K)-XFIN) 86,11,11
IF (X(K)-XFIN) 11,11,86
IF (Y(K)-3.00) 11,11,87
IF (10.0-Y(K)) 11,11,14
IF (YFIN-Y(1)) 90,21,89
IF (Y(K)-YFIN) 91,11,11
IF (Y(K)-YFIN) 11,11,91
IF (X(K)-3.00) 11,11,92
IF (16.0-X(K)) 11,11,14
N2 = K
T = TAU(K)
PRINT 13, N2
FORMAT (12H0 EARLY N2= 12/)
GO TO 81
CONTINUE
PRINT 16
FORMAT (1H06X2HN2 12X3HALF 14X1HT14X1HX13X1HY)
PRINT 17, N2, ALF, T, X(N2), Y(N2)
FORMAT (I10, 4F15.9 )
DELX = X(N2) - 26.0
DELY = Y(N2) + 1.0
EMFI = (973.75 + DELX*DELX + DELY*DELY)/1043.638 743
CAPV = VS(N2)*EMFI/18.702 181 818
XDOT = CAPV*COSP
YDOT = CAPV*SINP
FINJ = XJ(N2)
DETER = FINJ*(XDOT*XLAM + YDOT*XMU)
DIFX = XFIN - X(N2)
DIFY = YFIN - Y(N2)
DIFT = FINJ*(XLAM*DIFX + XMU*DIFY)/DETER
DIFA = (XDOT*DIFY - YDOT*DIFX)/DETER
T = T + FMUL*DIFT
ALF = ALF + FMUL*DIFA
PRINT 16
PRINT 17, N2, ALF, T, X(N2), Y(N2)
CONTINUE
PRINT 19
FORMAT(1H04X3HTAU8X1HX9X1HY6X6HCAPLAM5X2HWH8X2HWK8X2HVS9X1HS8X2HXJ
+8X2HPP8X2HQQ/)
PRINT 20, (TAU(I), X(I), Y(I), CAPLAM(I), WH(I), WK(I),
+ VS(I), S(I), XJ(I), PP(I), QQ(I), I=1, N2)
FORMAT (11F10.4)
STOP
END
SUBROUTINE ABC (H, A, B, C, DA, DB, DC)
RAD1 = SQRTF((0.038 635 06466*H-0.492 435 9727)*H+2.099 340 872)
A = -0.120 197 666*H + 20.748 910 236 - RAD1
RAD2 = SQRTF((0.032 937 36384*H-0.439 806 7919)*H+1.778 122 940)
B = -0.126 879 519*H + 20.633 462 763 - RAD2
RAD3 = SQRTF((0.010 441 8956*H-0.068 479 3891)*H+0.371 204 6341)
RAD4 = SQRTF((0.013 051 7910*H-0.193 791 6314)*H+0.814 886 3262)
C = 0.125 424 493*H - 0.293 444 8925 - RAD3 + RAD4
DA = -0.120 197 666 -(0.038 635 06466*H - 0.246 217 98635)/RAD1
DB = -0.126 879 519 -(0.032 937 36384*H - 0.219 903 39595)/RAD2
DC = 0.125 424 493 -(0.010 441 89560*H - 0.034 239 69455)/RAD3
+ (0.013 051 79100*H - 0.096 895 81570)/RAD4
END
SUBROUTINE POLAR (X, Y, P)
IF (X) 10, 3, 10
IF (Y) 8, 4, 6
P = 0.0
RETURN
P = 90.0
RETURN
P = -90.0

```

```

159
160
161
162
163
164
165
166
167
168
169
170
171
171.1
172
173 EW
173.1 E
173.2 W
173.3
173.4
174 NS
174.1 N
174.2 S
174.3
174.4
175
176
177
178
179
180
181
182
183
184
185
186
187
188
189
190
191
192
193
194
195
196
197
198
199
200
201
202
203
204
205
206
207
208
209
0
1
2
3
4
5
6
7
8
9
10
11
12
1
2
3
4
5
6
7
8

```



```

9 RETURN
0 P= 57.2957 7951 * ATANF(Y/X)
1 IF (X) 12,14,14
2 IF (Y) 15,13,13
3 P= P + 180.0
4 RETURN
5 P= P - 180.0
6 END
SUBROUTINE TERP (FUNC, X, Y, T, OUT, OUTX, OUTY)
DIMENSION FUNC(19,12,10),PT(4,4),P(4),Q(4),PX(4),QY(4),ORD(4),
+ ORDX(4),ORDY(4),S(4),D(4)
LL = 1
L = XINTF(T)
IF (L) 1,4,1
1 IF (8-L) 3,2,3
2 LL = 3
L = L-2
GO TO 4
3 LL = 2
L = L-1
4 M = XINTF(X) - 2
N = XINTF(Y) - 2
XX = 2.0*(X-INTF(X)) - 1.0
YY = 2.0*(Y-INTF(Y)) - 1.0
TT = 2.0*(T-INTF(T)) - 1.0
XP1= XX + 1.0
XM1= XX - 1.0
YP1= YY + 1.0
YM1= YY - 1.0
TP1= TT + 1.0
TM1= TT - 1.0
X2M= XP1*XM1
Y2M= YP1*YM1
T2M= TP1*TM1
P(1) = -XM1*X2M
P(2) = ((3.*XX+2.)*XX-9.)*XM1
P(3) = -2.*XX*XX + 18. - P(2)
P(4) = XP1*X2M
Q(1) = -YM1*Y2M
Q(2) = ((3.*YY+2.)*YY-9.)*YM1
Q(3) = -2.*YY*YY + 18. - Q(2)
Q(4) = YP1*Y2M
PX(4) = (3.*XX-1.)*XP1
PX(1) = 4.*XX - PX(4)
PX(2) = (9.*XX-11.)*XP1
PX(3) = -4.*XX - PX(2)
QY(4) = (3.*YY-1.)*YP1
QY(1) = 4.*YY - QY(4)
QY(2) = (9.*YY-11.)*YP1
QY(3) = -4.*YY - QY(2)
DO 6 K=1,4
DO 5 I=1,4
DO 5 J=1,4
5 PT(I, J) = FUNC(M+I,N+J,L+K)
DO 10 I=1,4
S(I) = P(1)*PT(1,I) + P(2)*PT(2,I) + P(3)*PT(3,I) + P(4)*PT(4,I)
0 D(I) = Q(1)*PT(I,1) + Q(2)*PT(I,2) + Q(3)*PT(I,3) + Q(4)*PT(I,4)
ORD(K) = (D(1)*P(1) + D(2)*P(2) + D(3)*P(3) + D(4)*P(4))/256.
ORDX(K) = (D(1)*PX(1) + D(2)*PX(2) + D(3)*PX(3) + D(4)*PX(4))/128.
6 ORDY(K) = (S(1)*QY(1) + S(2)*QY(2) + S(3)*QY(3) + S(4)*QY(4))/128.
IF (LL-2) 8,7,9
7 G = (3.0*TT+2.0)*TT - 9.0
H = G - 4.0*TT
OUT = (TM1*(G*ORD( 2)-T2M*ORD( 1))-TP1*(H*ORD( 3)-T2M*ORD( 4)))/16.
54 OUTX = (TM1*(G*ORDX( 2)-T2M*ORDX( 1))-TP1*(H*ORDX( 3)-T2M*ORDX( 4)))/16.
55 OUTY = (TM1*(G*ORDY( 2)-T2M*ORDY( 1))-TP1*(H*ORDY( 3)-T2M*ORDY( 4)))/16.
56 RETURN
57
8 TM3 = TT - 3.0
OUT = ( TM3*(TM1*ORD( 1)-2.0*TP1*ORD( 2)) + T2M*ORD( 3))/8.0
58
OUTX = ( TM3*(TM1*ORDX(1)-2.0*TP1*ORDX(2)) + T2M*ORDX(3))/8.0
59
OUTY = ( TM3*(TM1*ORDY(1)-2.0*TP1*ORDY(2)) + T2M*ORDY(3))/8.0
60
RETURN
61
9 TP3 = TT + 3.0
OUT = ( TP3*(TP1*ORD( 4)-2.0*TM1*ORD( 3)) + T2M*ORD( 2))/8.0
62
OUTX = ( TP3*(TP1*ORDX(4)-2.0*TM1*ORDX(3)) + T2M*ORDX(2))/8.0
63
OUTY = ( TP3*(TP1*ORDY(4)-2.0*TM1*ORDY(3)) + T2M*ORDY(2))/8.0
64
END
65
END
66
6 10 10 10 5-1.362675810 4.97704031150.00000000 0.500000000
67
PROGRAM STRAIGHT
68

```



```

TO CONVERT PROGRAM MINVOY TO PROGRAM STRAIGHT
(1) REPLACE MINVOY CARDS (43-55) BY FOLLOWING CARDS
COST = -SINK(KXST, KYST, 1)
OR COST = COSK(KXST, KYST, 1)
SINT = -COSK(KXST, KYST, 1)
OR SINT = -SINK(KXST, KYST, 1)
QUAD = BI*BI*COST*COST + AI*AI*SINT*SINT
RADI = SQRTF(QUAD - (CI*SINT)**2)
SINB = SINT*(AI*RADI - BI*CI*COST)/QUAD
COSB = (BI*COST*RADI + AI*CI*SINT*SINT)/QUAD
(2) REPLACE MINVOY CARDS (89-97) BY FOLLOWING CARDS
COST = -SK
OR COST = CK
SINT = -CK
OR SINT = -SK
QUAD = FB*FB*COST*COST + FA*FA*SINT*SINT
RADI = SQRTF(QUAD - (FC*SINT)**2)
SINB = SINT*(FA*RADI - FB*FC*COST)/QUAD
COSB = (FB*COST*RADI + FA*FC*SINT*SINT)/QUAD
(3) REPLACE MINVOY CARDS (153-161) BY FOLLOWING CARDS
COST = -SK
OR COST = CK
SINT = -CK
OR SINT = -SK
QUAD = FB*FB*COST*COST + FA*FA*SINT*SINT
RADI = SQRTF(QUAD - (FC*SINT)**2)
SINB = SINT*(FA*RADI - FB*FC*COST)/QUAD
COSB = (FB*COST*RADI + FA*FC*SINT*SINT)/QUAD
(4) REPLACE MINVOY CARDS (197-198) BY FOLLOWING CARDS
FF = (5.00 - Y(N2-1))/(Y(N2) - Y(N2-1))
OR FF = (13.0 - X(N2-1))/(X(N2) - X(N2-1))
T = FF*TAU(N2) + (1.0 - FF)*TAU(N2-1)

PROGRAM CONTOUR
DIMENSION ABS(900), ORD(900), KH(19, 12), HT(19, 12), KK(19, 12), PT(12),
+ COSK(19, 12), SINK(19, 12), RX(48), RY(48), CON(4, 4), IT(12), TI(12), D(3)
EQUIVALENCE (IT, TI), (LA, AL)
READ 1, DS, IHMAX, IDELH, IXMIN, IXMAX, IYMIN, IYMAX, KH, KK, TI, AL
1 FORMAT ( F4.2, 6I3/, 12(38I2/), 2(10A8/) )
DO 2 J=1, 12
DO 2 I=1, 19
2 HT(I, J) = KH(I, J)
PRINT 3, DS, IHMAX, IDELH, IXMIN, IXMAX, IYMIN, IYMAX
3 FORMAT ( 1H0, F4.2, 6I3 )
READ 26, (ABS(I), ORD(I), I=1, 5)
26 FORMAT (10F3.1)
CALL CRAW (5, ABS, ORD, 1, 0, LA, IT, 1., 1., 0, 0, 2, 2, 8, 8, 0, LAST)
C = DS*DS
DO 22 IY= IYMIN, IYMAX
Y = IY
DO 22 IX= IXMIN, IXMAX
X = IX
CALL COEF (HT, IX, IY, PT, CON)
DO 22 IH= 3, IHMAX, IDELH
H = IH
NR = 0
CALL ROOT (CON, 1, H, D, KER)
IF (KER) 8, 8, 4
4 DO 7 J=1, KER
ABS(1) = X + (D(J)+1.0)/2.0
ORD(1) = Y
IF (NR) 33, 35, 33
33 DO 34 I=1, NR
A = ABS(1) - RX(I)
B = ORD(1) - RY(I)
IF (A*A + B*B - C) 7, 7, 34
34 CONTINUE
35 CALL GRAD (PT, ABS(1), ORD(1), DX, DY, QUAD, DELX, DELY, DS, H, 0)
IF (QUAD - 1.E-10) 7, 7, 5
5 IF (DELY) 6, 36, 36
6 DS = -DS
DELX = DELX - DX - DX
DELY = DELY - DY - DY
IF (DELY) 7, 36, 36
36 ABS(2) = ABS(1) + DELX
ORD(2) = ORD(1) + DELY
CALL OUT (NR, ABS, ORD, RX, RY, DS, PT, X, Y, H)
7 CONTINUE
IF (KER) 13, 13, 9
8 CALL ROOT (CON, 2, H, D, KER)
9 DO 12 J=1, KER

```

```

43FS
43FE
44FS
44FE
45F
46F
47F
48F
89FS
89FE
90FS
90FE
91F
92F
93F
94F
153FS
153FE
154FS
154FE
155F
156F
157F
158F
197FS
197FE
198F
0
1
2
3
4
5
6
7
8
9
10
11
12
13
14
15
16
17
18
19
20
21
22
23
24
25
26
27
28
29
30
31
32
33
34
35
41
42
43
44
45
46
47
48
49
51
50
52

```



```

ABS(1) = X
ORD(1) = Y + (D(J)+1.0)/2.0
IF (NR) 37,39,37
37 DO 38 I=1,NR
A = ABS(1) - RX(I)
B = ORD(1) - RY(I)
IF (A*A + B*B - C) 12,12,38
38 CONTINUE
39 CALL GRAD (PT,ABS(1),ORD(1),DX,DY,QUAD,DELX,DELY,DS,H,0)
IF (QUAD - 1.E-10) 12,12,10
10 IF (DELX) 11,40,40
11 DS = -DS
DELY= DELY - DY - DY
DELX= DELX - DX - DX
IF (DELX) 12,40,40
40 ABS(2) = ABS(1) + DELX
ORD(2) = ORD(1) + DELY
CALL CUT (NR,ABS,ORD,RX,RY,DS,PT,X,Y,H)
12 CONTINUE
13 CALL ROOT (CON, 3, H, D, KER)
IF (KER) 18,18,14
14 DO 17 J=1,KER
ABS(1) = X + (D(J)+1.0)/2.0
ORD(1) = Y + 1.0
IF (NR) 41,43,41
41 DO 42 I=1,NR
A = ABS(1) - RX(I)
B = ORD(1) - RY(I)
IF (A*A + B*B - C) 17,17,42
42 CONTINUE
43 CALL GRAD (PT,ABS(1),ORD(1),DX,DY,QUAD,DELX,DELY,DS,H,0)
IF (QUAD - 1.E-10) 17,17,15
15 IF (DELY) 44,44,16
16 DS = -DS
DELX= DELX - DX - DX
DELY= DELY - DY - DY
IF (DELY) 44,44,17
44 ABS(2) = ABS(1) + DELX
ORD(2) = ORD(1) + DELY
CALL CUT (NR,ABS,ORD,RX,RY,DS,PT,X,Y,H)
17 CONTINUE
18 CALL ROOT (CON, 4, H, D, KER)
IF (KER) 22,22,19
19 DO 22 J=1,KER
ABS(1) = X + 1.0
ORD(1) = Y + (D(J)+1.0)/2.0
IF (NR) 45,47,45
45 DO 46 I=1,NR
A = ABS(1) - RX(I)
B = ORD(1) - RY(I)
IF (A*A + B*B - C) 22,22,46
46 CONTINUE
47 CALL GRAD (PT,ABS(1),ORD(1),DX,DY,QUAD,DELX,DELY,DS,H,0)
IF (QUAD - 1.E-10) 22,22,20
20 IF (DELX) 48,48,21
21 DS = -DS
DELY= DELY - DY - DY
DELX= DELX - DX - DX
IF (DELX) 48,48,22
48 ABS(2) = ABS(1) + DELX
ORD(2) = ORD(1) + DELY
CALL CUT (NR,ABS,ORD,RX,RY,DS,PT,X,Y,H)
22 CONTINUE
DO 23 I=1,19
DELI = I-26
DO 23 J=1,12
DELJ = J+1
RAD = SQRTF(DELI*DELI + DELJ*DELJ)
ANGLE = KK(I,J)*10
ANGLE = ANGLE/57.2957 7951
COS = COSF(ANGLE)
SIN = SINF(ANGLE)
COSK(I,J) = -(DELI*COS + DELJ*SIN)/RAD
23 SINK(I,J) = (DELI*SIN - DELJ*COS)/RAD
IXP = IXMIN + 1
IYP = IYMIN + 1
DO 24 IY = IYP, IYMAX
Y = IY
DO 24 IX = IXP, IXMAX
X = IX
COS = COSK(IX,IY)

```

53
54
55
56
57
58
59
60
61
62
68
69
70
71
72
73
74
75
76
77
78
79
80
81
82
83
84
85
86
87
88
89
95
96
97
98
99
100
101
102
103
104
105
106
107
108
109
110
111
112
113
114
115
116
122
123
124
125
126
127
128
129
130
103
104
105
106
107
108
109
110
111
112
113
114
116
118
119
120
121
122

	SIN = SINK(IX,IY)	123
	ABS(1) = X + COS/3.0	124
	ORD(1) = Y + SIN/3.0	125
	ABS(2) = X + X - ABS(1)	126
	ORD(2) = Y + Y - ORD(1)	127
	ABS(3) = ABS(2) + (1.732050807*COS - SIN)/18.0	128
	ORD(3) = ORD(2) + (1.732050807*SIN + COS)/18.0	129
	ABS(4) = ABS(3) + SIN/9.0	130
	ORD(4) = ORD(3) - COS/9.0	131
	ABS(5) = ABS(2)	132
	ORD(5) = ORD(2)	133
	DO 28 I=1,5	134
	ABS(I) = 0.8*ABS(I) - 4.0	135
28	ORD(I) = 0.8*ORD(I) - 3.2	136
	CALL CRAW (5,ABS,ORD,2,0,LA,IT,1.,1.,0,0,2,2,8,8,0, LAST)	137
24	CONTINUE	139
	READ 29, (ABS(I), I=1,52)	140
29	FORMAT (4(13F6.3/))	141
	READ 29, (ORD(I), I=1,52)	142
	DO 30 I=1,52	143
	ABS(I) = 0.8*ABS(I) - 4.0	144
30	ORD(I) = 0.8*ORD(I) - 3.2	145
	CALL CRAW (52,ABS,ORD,2,0,LA,IT,1.,1.,0,0,2,2,8,8,0, LAST)	146
	READ 29, (ABS(I), I=1,52)	148
	READ 29, (ORD(I), I=1,52)	149
	DO 32 I=1,52	150
	ABS(I) = 0.8*ABS(I) - 4.0	151
32	ORD(I) = 0.8*ORD(I) - 3.2	152
	CALL CRAW (52,ABS,ORD,3,0,LA,IT,1.,1.,0,0,2,2,8,8,0, LAST)	153
25	STOP 25	154
	END	155
	SUBROUTINE COEF (HT, IX, IY, P, CON)	0
	DIMENSION HT(19,12), P(12), CON(4,4)	1
	P(1) = HT(IX,IY-1)	2
	P(2) = HT(IX+1,IY-1)	3
	LX = IX-2	4
	DO 1 I=1,4	5
	P(2+I) = HT(LX+I,IY)	6
1	P(6+I) = HT(LX+I,IY+1)	7
	P(11) = HT(IX,IY+2)	8
	P(12) = HT(IX+1,IY+2)	9
	CON(1,1) = P(6) - P(3) + 3.0*(P(4)-P(5))	10
	CON(2,1) = P(3) - P(4) - P(5) + P(6)	11
	CON(3,1) = P(3) - P(6) + 11.*(P(5)-P(4))	12
	CON(4,1) = -P(3) - P(6) + 9.0*(P(5)+P(4))	13
	CON(1,3) = P(10) - P(7) + 3.0*(P(8)-P(9))	14
	CON(2,3) = P(10) - P(9) - P(8) + P(7)	15
	CON(3,3) = P(7) - P(10) + 11.*(P(9)-P(8))	16
	CON(4,3) = -P(7) - P(10) + 9.0*(P(9)+P(8))	17
	CON(1,2) = P(11) - P(1) + 3.0*(P(4)-P(8))	18
	CON(2,2) = P(1) - P(4) - P(8) + P(11)	19
	CON(3,2) = P(1) - P(11) + 11.*(P(8)-P(4))	20
	CON(4,2) = -P(1) - P(11) + 9.0*(P(8)+P(4))	21
	CON(1,4) = P(12) - P(2) + 3.0*(P(5)-P(9))	22
	CON(2,4) = P(12) - P(9) - P(5) + P(2)	23
	CON(3,4) = P(2) - P(12) + 11.*(P(9)-P(5))	24
	CON(4,4) = -P(2) - P(12) + 9.0*(P(9)+P(5))	25
	END	26
	SUBROUTINE ROOT (CON, KS, H, D, KER)	0
	DIMENSION CON(4,4), D(3), X(3)	1
	FOURTH = CON(4,KS) - 16.0*H	2
	IF (CON(1,KS)) 12,2,12	5
2	IF (CON(2,KS)) 6,3,6	6
3	IF (CON(3,KS)) 11,4,11	7
4	IF (FOURTH) 9,5,9	8
5	D(1) = -1.0	9
	D(2) = 1.0	10
	KER = 2	11
	RETURN	12
6	B = 0.5*CON(3,KS)/CON(2,KS)	13
	C = FOURTH/CON(2,KS)	14
	QUAD = B*B - C	15
	RAD = SQRTF(ABS(QUAD))	16
	IF (QUAD) 7,8,10	17
7	IF (RAD - 1.E-5) 8,8,9	18
8	X(1) = -B	19
	KER = 1	20
	GO TO 28	21
9	KER = 0	22
	RETURN	23
10	X(1) = -B + RAD	24


```

X(2) = -B - RAD
KER = 2
GO TO 28
11 X(1) = -FOURTH/CON(3,KS)
KER = 1
GO TO 28
12 A = CON(2,KS)/CON(1,KS)
B = CON(3,KS)/CON(1,KS)
C = FOURTH/CON(1,KS)
P = A*A/9.0 - B/3.0
Q = A*B/6.0 - C/2.0 - A*A*A/27.0
RAD = SQRTF(ABSF(P))
IF (P) 13,16,18
13 PHI3 = ASINH(Q/(RAD*RAD*RAD))/3.0
X(1) = 2.0*RAD*SINH(PHI3) - A/3.0
KER = 1
IF (1.732050807*RAD*COSH(PHI3) - 1.E-5) 15,15,28
15 X(2) = -(X(1)+A)/2.0
KER = 2
GO TO 28
16 X(1) = (2.0*ABSF(Q)**.3333333333333333 - A/3.0
KER = 1
IF (Q) 17,28,28
17 X(1) = -X(1) - A/1.5
GO TO 28
18 ARG = Q/(RAD*RAD*RAD)
IF (ABSF(ARG) - 1.0) 27,25,19
19 PHI3 = ACOSH(ABSF(ARG))/3.0
IF (Q) 21,24,24
21 X(1) = -2.0*RAD*COSH(PHI3) - A/3.0
22 KER = 1
IF (1.732050807*RAD*SINH(PHI3) - 1.E-5) 15,15,28
24 X(1) = 2.0*RAD*COSH(PHI3) - A/3.0
GO TO 22
25 X(1) = 2.0*ABSF(Q)**.3333333333333333 - A/3.0
X(2) = -(X(1)+A)/2.0
KER = 2
IF (Q) 26,28,28
26 X(2) = -X(2) - A/1.5
GO TO 17
27 PHI3 = ACOSF(ARG)/3.0
SIN = 1.732050807*RAD*SINF(PHI3)
Z = RAD*COSF(PHI3)
X(1) = Z + Z - A/3.0
X(2) = -Z + SIN - A/3.0
X(3) = X(2) - SIN - SIN
KER = 3
28 JAR = 0
DO 30 I=1,KER
IF (ABSF(X(I)) - 1.0) 29,29,30
29 JAR = JAR + 1
D(JAR) = X(I)
30 CONTINUE
KER = JAR
END
SUBROUTINE GRAD (P, X, Y, DX, DY, QUAD, DELX, DELY, DS, H, KD)
DIMENSION P(12)
XX = 2.0*(X - INTF(X)) - 1.0
YY = 2.0*(Y - INTF(Y)) - 1.0
XP1= XX + 1.0
XM1= XX - 1.0
YP1= YY + 1.0
YM1= YY - 1.0
TX= 2.0*XX
TY= 2.0*YY
CIR = 3.0*(XX*XX+YY*YY) - 10.0
X2M = XP1*XM1
Y2M = YP1*YM1
SXP = 6.0*XX + 2.0
SXM = SXP - 4.0
SYP = 6.0*YY + 2.0
SYM = SYP - 4.0
QXP = 18.0*XX + 2.0
QXM = QXP - 4.0
QYP = 18.0*YY + 2.0
QYM = QYP - 4.0
Q1 = X2M*P(10) + Y2M*P(12) - (CIR-TX-TY)*P(9)
Q2 = X2M*P( 7) + Y2M*P(11) - (CIR+TX-TY)*P(8)
Q3 = X2M*P( 3) + Y2M*P( 1) - (CIR+TX+TY)*P(4)
Q4 = X2M*P( 6) + Y2M*P( 2) - (CIR-TX+TY)*P(5)
P1 = (TX*P(10) - SXM*P(9))*XP1

```



```

P2 = (TX*P( 7) - SXP*P(8))*XM1
P3 = (TX*P( 3) - SXP*P(4))*XM1
P4 = (TX*P( 6) - SXM*P(5))*XP1
P5 = (TY*P(12) - SYM*P(9))*YP1
P6 = (TY*P( 2) - SYP*P(5))*YM1
P7 = (TY*P( 1) - SYP*P(4))*YM1
P8 = (TY*P(11) - SYM*P(8))*YP1
HX = (YP1*(Q1-Q2+P1-P2) + YM1*(Q3-Q4+P3-P4))/16.0
HY = (XP1*(Q1-Q4+P5-P6) + XM1*(Q3-Q2+P7-P8))/16.0
QUAD = HX*HX + HY*HY
IF (QUAD - 1.E-10) 3,3,2
2 IF (KC) 5,4,5
4 HXX = (YP1*(SXP*P(10)-QXP*P(9)-SXM*P(7)+QXM*P(8)) +
+ YM1*(SXM*P( 3)-QXM*P(4)-SXP*P(6)+QXP*P(5)))/8.0
39
+ HYY = (XP1*(SYP*P(12)-QYP*P(9)-SYM*P(2)+QYM*P(5)) +
40
+ XM1*(SYM*P( 1)-QYM*P(4)-SYP*P(11)+QYP*P(8)))/8.0
41
HXY = (P1-P4+P3-P2+P5-P8+P7-P6+Q1-Q2+Q3-Q4)/4.0
42
RAD = SQRTF(QUAD)
43
DX = DS*HY/RAD
44
DY = -DS*HX/RAD
45
SEC = 0.5*(HXX*DX*DX + HXY*DX*DY + HYY*DY*DY)/QUAD
46
DELX= DX - HX*SEC
47
DELY= DY - HY*SEC
48
RETURN
49
5 DH = H - (YP1*(XP1*Q1-XM1*Q2) + YM1*(XM1*Q3-XP1*Q4))/32.0
50
DX = CH*HX/QUAD
51
DY = CH*HY/QUAD
52
DELX = DH*DH/QUAD
53
3 END
54
SUBROUTINE OUT (NR,ABS,ORD,RX,RY,DS,PT,X,Y,H)
0
DIMENSION ABS(900),ORD(900),RX(48),RY(48),PT(12)
1
C = 2.0*DS*DS
1.5
NPT = 1
2
NR = NR + 1
3
RX(NR) = ABS(1)
4
RY(NR) = ORD(1)
5
GO TO 1
6
4 IF (40-NPT) 20,20,24
7
24 CALL GRAD (PT,ABS(NPT),ORD(NPT),DX,DY,QUAD,DELX,DELY,DS,H,0)
8
IF (QUAD - 1.E-10) 19,19,5
9
5 ABS(NPT+1) = ABS(NPT) + DELX
14
ORD(NPT+1) = ORD(NPT) + DELY
15
1 CALL GRAD(PT,ABS(NPT+1),ORD(NPT+1),DX,DY,QUAD,DELX,DELY,DS,H,1)
16
IF (QUAD - 1.E-10) 19,19,6
17
6 IF (DELX - C) 25,25,19
18
25 NPT = NPT + 1
19
ABS(NPT) = ABS(NPT) + DX
20
ORD(NPT) = ORD(NPT) + DY
20.1
NPT1= NPT
21
IF (ORD(NPT)-Y) 7,10,8
22
7 NPT1= NPT1 - 1
23
GO TO 10
24
8 IF (Y+1.0-ORD(NPT)) 9,10,10
25
9 NPT1= NPT1 - 1
26
10 IF (ABS(NPT1)-X) 11,14,12
27
11 NPT1= NPT1 - 1
28
GO TO 15
29
12 IF (X+1.0-ABS(NPT1)) 11,14,14
30
14 IF (NPT-NPT1) 15,4,15
31
15 NPT = NPT1
32
19 IF (NPT-1) 26,26,20
33
20 NR = NR + 1
34
RX(NR) = ABS(NPT)
35
RY(NR) = ORD(NPT)
36
DO 21 I=1,NPT
37
ABS(I) = 0.8*ABS(I) - 4.0
38
ORD(I) = 0.8*ORD(I) - 3.2
39
LA = 4H
40
CALL CRAW (NPT,ABS,ORD,2,0,LA,7,1.,1.,0,0,2,2,8,8,0,LA$T)
41
RETURN
42
26 NR = NR - 1
43
END
44
END
45
0 21 3 6 13 4 10 ZEROTH

```

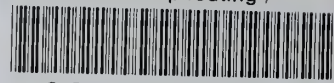

TA7 74564
.U6 Bleick
no.46 Minimal-time ship
routing.

7 AUG 68	17289
7 FEB 69	17980
27 NOV 70	18919
6 DEC 71	20553
13 JUN 74	S10519

TA7 74564
.U6 Bleick
no.46 Minimal-time ship
routing.

genTA 7.U6 no.46

Minimal-time ship routing /



3 2768 001 61489 4
DUDLEY KNOX LIBRARY